

SFA Modernization Program
United States Department of Education
Student Financial Assistance



Campus-Based Programs
Detailed Logical Design

Task Order #49
Deliverable #49.1.3

Table Of Contents

1. INTRODUCTION.....	3
1.1 PURPOSE.....	3
1.2 SCOPE.....	3
2. SYSTEM OVERVIEW	4
2.1 RE-DESIGN CAMPUS-BASED SYSTEM.....	4
2.2 DOCUMENT OVERVIEW.....	4
3. LOGICAL ARCHITECTURE.....	5
3.1 OVERVIEW	5
3.2 WEB.....	6
3.2.1 Overview.....	6
3.2.2 Classes	7
3.3 ADMIN.....	7
3.3.1 Overview.....	7
3.3.2 Classes	8
3.4 REPORTS.....	8
3.4.1 Overview.....	8
3.4.2 Classes	8
3.5 DATABASE INTERFACE	8
3.5.1 Overview.....	8
3.5.2 Classes	8
3.6 ORACLE DATABASE	9
3.6.1 Overview.....	9
3.6.2 Classes	9
3.7 FISAP FORM.....	9
3.7.1 Overview.....	9
3.7.2 Classes	9

1 Introduction

1.1 *Purpose*

The purpose of this document is to give an overview of the class structure and packages that will be the basis of the Campus based modernization. The class structure includes class relationships, operations, and attributes. The interactions between the objects, created from the classes, are detailed in the sequence diagrams.

1.2 *Scope*

This document gives an overview of the software architecture of the Campus Based Modernization. This document is a living document that over the life of the project will be updated as the design is developed. The development of the design will be driven by the use cases. The use case will be used to develop the sequence diagrams and add/remove classes and objects as necessary to support the sequences. The sequences will also help define the class relationships, operations, and attributes.

This document is generated as a SODA report from Rational Rose. Each generation of this document will cause an increment of the revision number in this document.

2 System Overview

2.1 *Re-design Campus-Based System*

The scope of the system includes the FISAP entry, FISAP data collection, Campus Based Administration, replacement of CB Windows, and development of the Oracle database. There are three major interfaces to the system. The interfaces are the institution user interface, web administration user interface (WAUI), and the database interface. The institution user interface is a web based interface, which is used by the institution to enter data into the FISAP sections, validation check the FISAP form and submit the FISAP form. The WAUI is used by the Department Of Education's campus based administration staff to administer the submitted FISAP forms, edit data as necessary, interface with accounting, view various reports and allocate funds to the institutions participating in the program.

2.2 *Document Overview*

The current version of this document has the logical view of classes and their documentation from the Rose model. Currently no sequence diagrams have been developed. As the use cases for the WAUI features are developed the sequence diagrams for this design will be developed.

The logical architecture starts with an overview of the entire system. This consists of a diagram with all of the major classes and their relationships shown. After the overview are the packages. Packages are the logical grouping of classes. Currently the packages are:

- WEB, Institution and WAUI web interface.
- Admin, Department of education classes that implement the required features.
- Reports, Functionality to develop the reports needed by the system users.
- Database Interface, Database features needed to support the system.
- Oracle database, Relational database used by the system.
- FISAP Form, Classes that realize the features necessary for institution use of the FISAP Form.

3 LOGICAL ARCHITECTURE

The following diagrams are a first try at the architecture of the Campus Based Modernization. There are three stereotypes used in this architecture. They are boundary, entity, and control. The boundary is shows forms, obtains user input and interfaces to the database. The control class provides the communication between the boundary and entity classes. the entity classes add a level of abstraction to the architecture. The abstraction is based on design patterns that encourage reuse. With this level of abstraction if a boundary class (user interface changes it does not affect the entity classes. The third class stereotype is the entity classes. These hold the major data and functionality of the system. As the sequence diagrams are detailed this diagram will change. Classes may be added or deleted. Attributes, operations and relationships will be detailed, added and deleted.

3.1 Overview

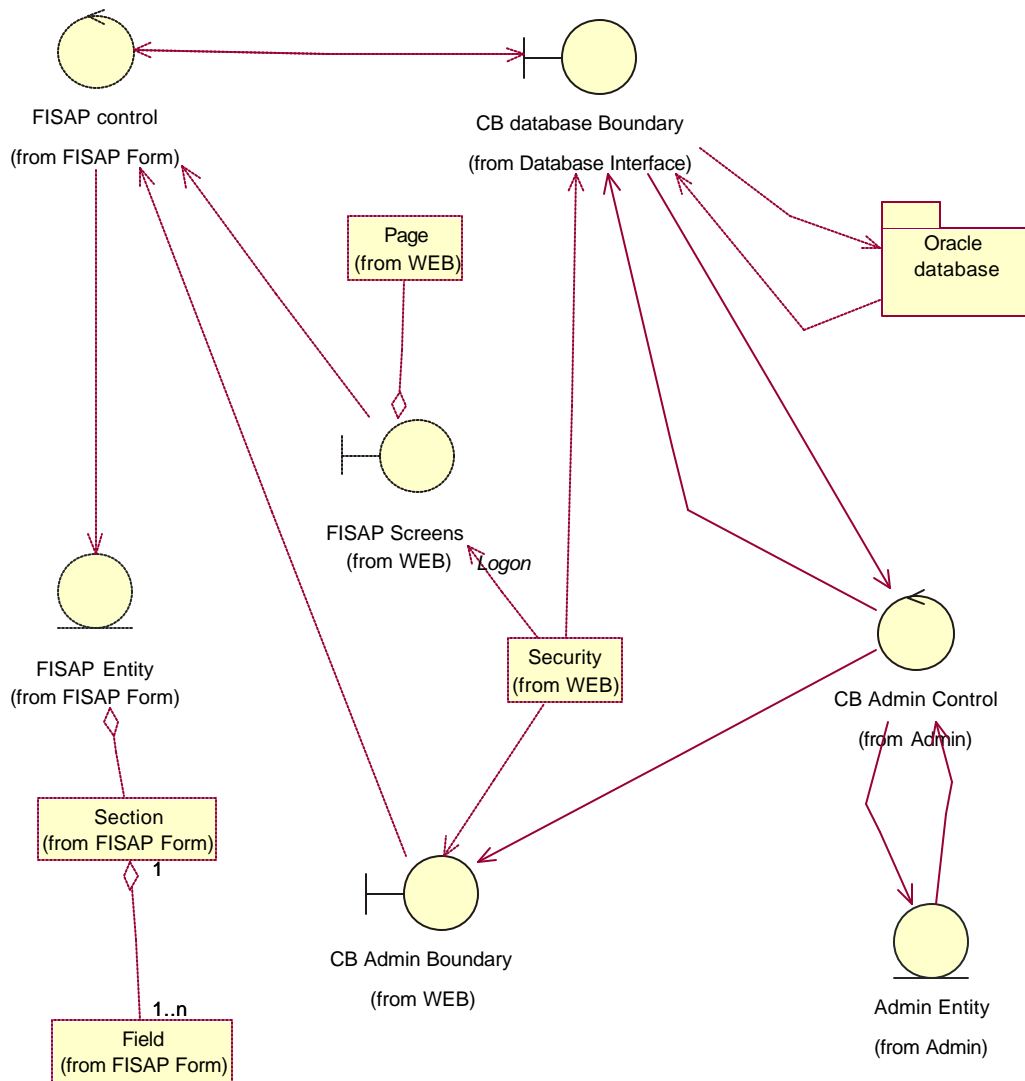


Figure 1: [Proposed Architecture1](#)

3.2 WEB

3.2.1 Overview

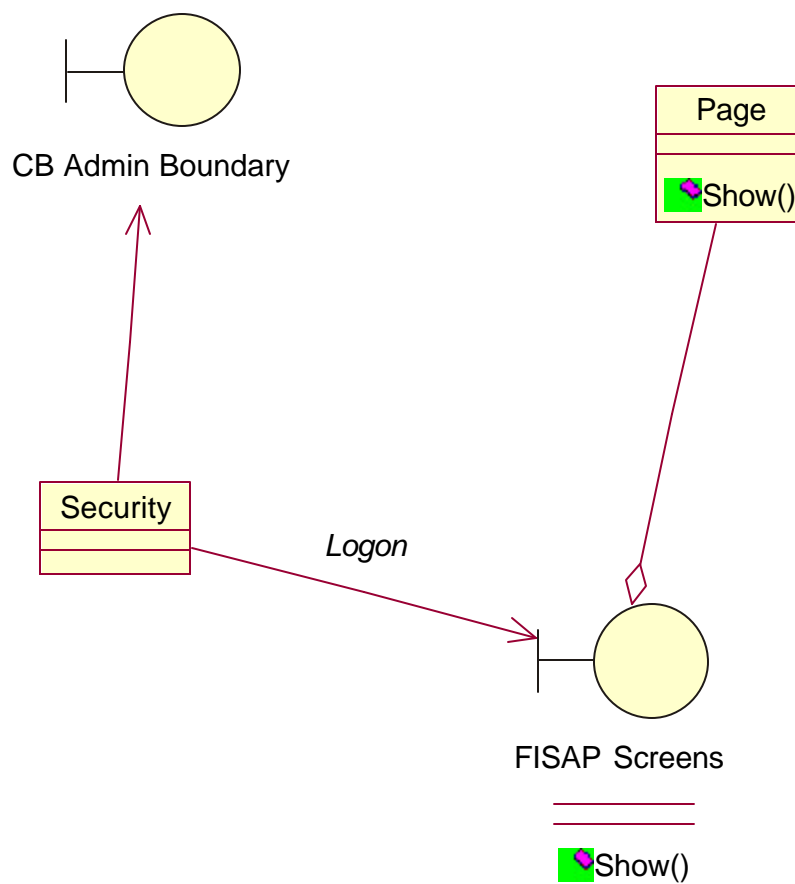


Figure 2: Main

3.2.2 Classes

3.2.2.1 FISAP Screens (NormalClass)

Boundary class. This class displays the FISAP Screens and captures user entry data.

3.2.2.2 Security (NormalClass)

A user must get past the security to access the boundary classes

3.2.2.3 CB Admin Boundary (NormalClass)

The user interface for the campus based administrators.

3.2.2.4 Page (NormalClass)

The FISAP document contains many pages.

3.3 Admin

3.3.1 Overview

Administrative operations including FISAP editing, accounting interfaces, and database reports.

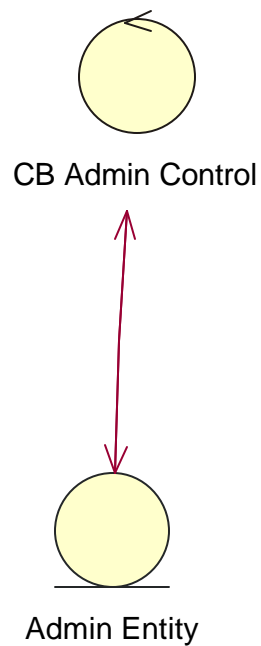


Figure 3: [Main](#)

3.3.2 Classes

3.3.2.1 Admin Entity (NormalClass)

3.3.2.2 CB Admin Control (NormalClass)

Communication module for the CB Administration. Partial replacement for the CB windows module.

3.4 Reports

3.4.1 Overview

3.4.2 Classes

3.5 Database Interface

3.5.1 Overview

Relational Database with CB windows as the boundary class

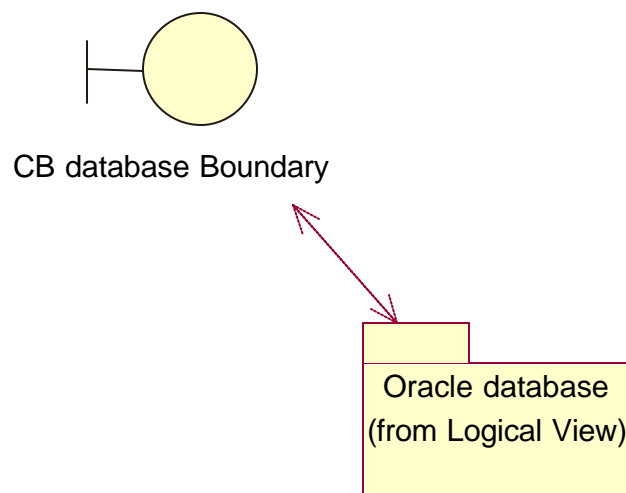


Figure 4: [Main](#)

3.5.2 Classes

3.5.2.1 CB database Boundary (NormalClass)

Interface to the database. Reads and writes data from the database.

3.6 Oracle database

3.6.1 Overview

Relational database. The ER diagram is not included in this design.

3.6.2 Classes

3.7 FISAP Form

3.7.1 Overview

Contains the FISAP operations and attributes.

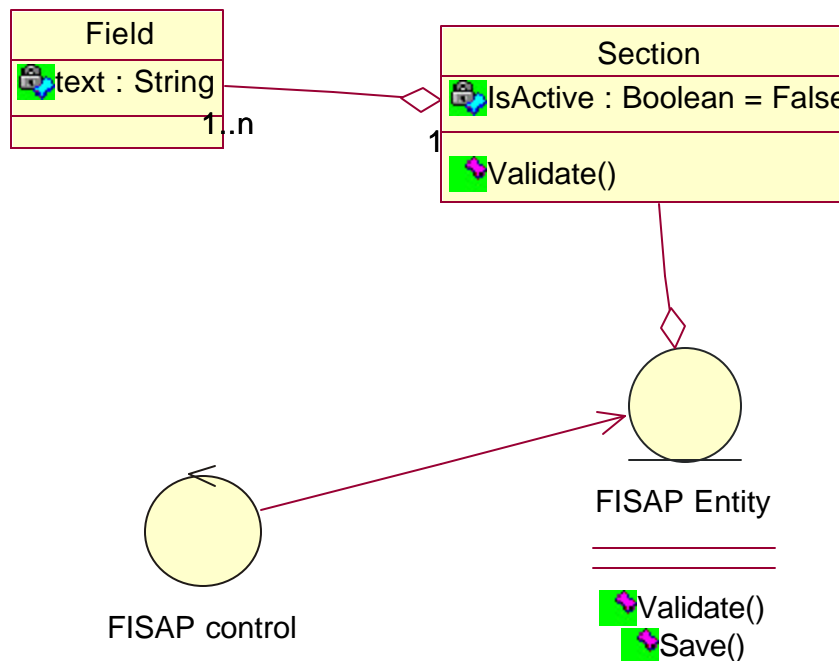


Figure 5: FISAP Main

3.7.2 Classes

3.7.2.1 Field (NormalClass)

Contains operations and attributes needed for field data collection. Certain fields are display only. Other fields can contain only certain characters.

3.7.2.2 Section (NormalClass)

Each section contains many fields. Most validation edit checks are section specific.

3.7.2.3 FISAP control (NormalClass)

Communication coordinator.

3.7.2.4 FISAP Entity (NormalClass)

Holds the functionality and data for the FISAP form. Contains many sections. The validation checks will be spread across the sections